Continuation semantics ii[1]

Patrick D. Elliott[2] *& Martin Hackl*[3]

February 13, 2020

[2] pdell@mit.edu

[3] hackl@mit.edu

> **Homework**
> -
>
> - Finish reading Barker & Shan 2014 chapters 1 and 4, if you haven't already.
>
> - Read Barker & Shan 2014 chapter 7.
>
> - Do p-set 2 (to be posted on Stellar later today)!

1   Roadmap

This week, I'll finish introducing *continuation semantics*. We'll minimally try to cover:

- The syntax-semantics interface (more explicitly, this time).

- Inverse scope via multi-story towers.

- Split scope.

- Scope islands as *evaluation islands* + remarks on scope economy.

- Generalized con/dis-junction + "split scope" readings.

With potentially two extensions:

- DP-internal composition and indexed continuations.

- Exceptionally-scoping indefinites via continuations (Charlow 2014).

2   A note on syntax

So far, I've been a little shy about saying explicitly what we're assuming here about syntax, and what we're assuming about the syntax-semantics mapping.

I'll assume a derivational theory, according to which structures are built-up via successive application of MERGE.[4]

(1)   MERGE
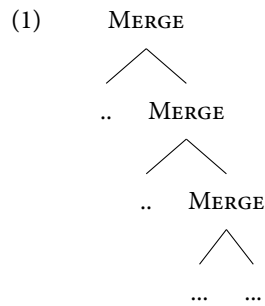
I'll furthermore adopt the hypothesis that the syntactic derivation proceeds in lockstep with the semantic computation. This conjecture, which goes back at least to Montague (1973), is often described as *direct compositionality*.[5]

Minimally, the formatives must be *tuples* consisting of phonological features and semantic features: (phon, sem, ...). Semantic features could be cashed out as model theoretic objects, or perhaps as expressions of the simply typed lambda calculus.

I'll assume that part of what MERGE does is concatenate phonological features. This is because MERGE is just an instruction for combining formatives. On the semantic side, it typically does function application.[6]

(2) $(\mathbb{x}, x) * (\mathbb{y}, y) := ([\mathbb{x} \ \mathbb{y}], x \ \mathsf{A} \ y)$

It can also do concatenation of phonological features, plus Scopal Function Application (SFA) of semantic values (whence the left-to-right bias of S).

(3) $(\mathbb{x}, x) * (\mathbb{y}, y) := ([\mathbb{x} \ \mathbb{y}], x \ \mathsf{S} \ y)$

I've define LIFT as a purely semantic operation – this is to be taken as shorthand for an operation on a formative that only effects the semantic value:

(4) $(\mathbb{x}, x)^{\uparrow} := (\mathbb{x}, x^{\uparrow})$

This constitutes the basics of the system laid out in Elliott 2019. See Elliott for an elaboration of how to supplement this system with a feature calculus, but for today's purposes we won't need any additional assumptions.

[4] I'll often use "syntactic structure speak" when talking about trees. This is harmless, since they can always be interpreted as the graph of a syntactic derivation, especially since trees encode both structure and order.

[5] Although direct compositionality is often associated with frameworks such as variable free semantics and Combinatory Categorial Grammar, it's in principle independent. See, e.g., Kobele (2006) for an explicit formalization of a directly compositional minimalist grammar.

[6] It follows that MERGE is a *non-symmetric relation*, departing from, e.g., Chomsky (1995), but consistent with Stabler (1997) and related work.

2.1   *Deriving inverse scope*

Last type we got as far as being able to derive *surface scope* readings, as well as scopal ambiguities arising via interactions with scopally-immobile expressions.

We achieved this latter coverage by allowing LOWER to fix the scope of a quantifier at different points in the derivation.

We still don't have any way of accounting for scopal interactions between multiple scopally-mobile expressions (i.e., quantifiers). Since every major theory of quantifier scope is tailored to achieve this, we have a major problem on our hands!

Fortunately, we already have all of the primitive operations we need in order to achieve inverse scope readings.

Recall that LIFT is a *polymorphic function* – it lifts a value into a trivial tower:

(5) $a^{\uparrow} := \dfrac{[\,]}{a}$

Since LIFT is polymorphic, in principle it can apply to any kind of value – even a tower! Let's flip back to lambda notation to see what happens.

(6) $[\![\text{everyone}]\!] := \lambda k \,.\, \forall x[k\ x]$ $\qquad\qquad\qquad$ $(e \to t) \to t$

(7) $[\![\text{everyone}]\!]^{\uparrow} = \lambda l \,.\, l\,(\lambda k \,.\, \forall x[k\ x])$ $\qquad$ $(((e \to t) \to t) \to t) \to t$

Going back to tower notation, lifting a tower adds a trivial third story:[7] Following Charlow (2014), when we apply LIFT to a tower, we'll describe the operation as *external lift* (although, it's worth bearing in mind that this is really just our original LIFT function).

The third story of the tower corresponds to whatever takes scope over the outer continuation variable (here, l), and the second story of the tower corresponds to whatever takes scope over the inner continuation variable (here, k).

(8) $\left(\dfrac{\forall x[\,]}{x}\right)^{\uparrow} = \dfrac{\dfrac{[\,]}{\forall x[\,]}}{x}$

[7] In fact, via successive application of LIFT, we can generate an $n-$story tower.

One important thing to note is that, when we externally lift a tower, the quantificational part of the meaning always remains on the same story relative to the bottom story. Intuitively, this reflects the fact that, ultimately, LIFT alone isn't going to be enough to derive quantifier scope ambiguities.

> **Question**
> -
> Which (if any) of the following bracketings make sense for a three-story tower:
>
> (9) $\dfrac{\left(\dfrac{f\,[]}{g\,[]}\right)}{x}$
>
> (10) $\dfrac{\dfrac{f\,[]}{}}{\left(\dfrac{g\,[]}{x}\right)}$

The extra ingredient we're going to need in order to derive inverse scope, is the ability to sandwhich an empty story into the *middle* of our tower, pushing the quantificational part of the meaning to the very top.

This is *internal lift* ($\Uparrow$).[8]

(12) *Internal lift* (def.)
    a.   ($\Uparrow$) : $K_t\ a \to K_t\ (K_t\ a)$
    b. $m^{\Uparrow} := \lambda k \,.\, m\,(\lambda x \,.\, k\,x^{\uparrow})$

It's much easier to see what internal lift is doing by using the tower notation. We can also handily compare its effects to those of *external* lift.

(13) *Internal lift* (tower ver.)

$$\left(\dfrac{f\,[]}{x}\right)^{\Uparrow} := \dfrac{\dfrac{f\,[]}{[]}}{x}$$

(14) *External lift* (tower ver.)

$$\left(\dfrac{f\,[]}{x}\right)^{\uparrow} := \dfrac{\dfrac{[]}{f\,[]}}{x}$$

Armed with *internal* and *external* lifting operations, we now have everything we need to derive inverse scope. We'll start with a simple example (15).

The trick is: we *internally* lift the quantifier that is destined to take wide scope.

---

[8] I can tell what you're thinking: "seriously? Another *darn* type-shifter? How many of these are we going to need?!". Don't worry, I got you. Even thought we've defined internal lift here as a primitive operation, it actually just follows from our existing machinery. Concretely, *internal lift* is really just *lifted* LIFT (so many lifts!). Lifted LIFT applies to its argument via S.

(11) $\dfrac{[]}{\uparrow}\,S\,\dfrac{f\,[]}{x} = \dfrac{\dfrac{f\,[]}{[]}}{x}$

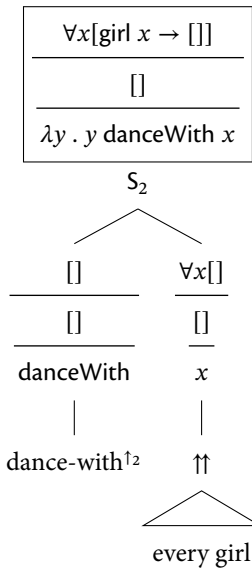(15)   A boy danced with every girl. $\forall > \exists$

Before we proceed, we need to generalize LIFT and SFA to three-story towers.[9]
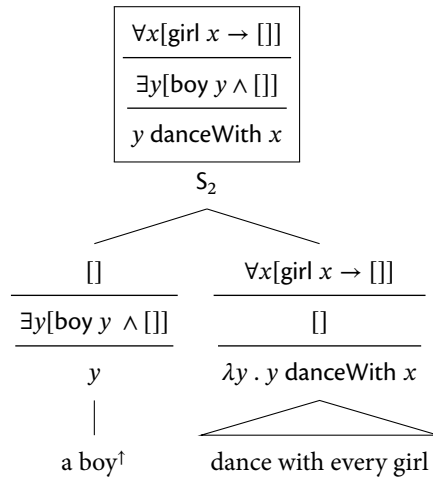
(16) $x^{\uparrow 2} := \dfrac{\dfrac{[]}{[]}}{x}$

(17) $\dfrac{f\,[]}{m}\,\mathsf{S}_2\,\dfrac{g\,[]}{n} := \dfrac{f\,(g\,[])}{m\,\mathsf{S}\,n}$

(18)   Step 1: internally lift *every girl*

(19)   Step 2: *ex*ternally lift *a boy*

What we're left with now is a 3-story tower with the universal on the top story and the existential on the middle story. We can collapse the tower by first collapsing the bottom two stories, and then collapsing the result. In order to do this, we'll first define *internal lower*.[10]

(21)   *Internal lower* (def)
   a. $(\Downarrow) : K_t\,(K_t\,a) \rightarrow K_t\,a$
   b. $m^{\Downarrow} := \lambda k\,.\,m\,(\lambda n\,.\,k\,n^{\downarrow})$

(22)   *Internal lower* (def.) $\left(\dfrac{\dfrac{f\,[]}{g\,[]}}{p}\right)^{\Downarrow} := \dfrac{f\,[]}{\left(\dfrac{g\,[]}{p}\right)^{\downarrow}}$

Now we can collapse the tower by doing *internal lower*, followed by *lower*:

(23) $\boxed{\forall x[\text{girl } x \rightarrow (\exists y[\text{boy } y \wedge y \text{ danceWith } x])]}$

|

$\downarrow$

|

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\exists x[\text{boy } x \wedge y \text{ danceWith } x]}$$

|

$\Downarrow$

|

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\dfrac{\exists y[\text{boy } y \wedge []]}{y \text{ danceWith } x}}$$

a boy danced with every girl
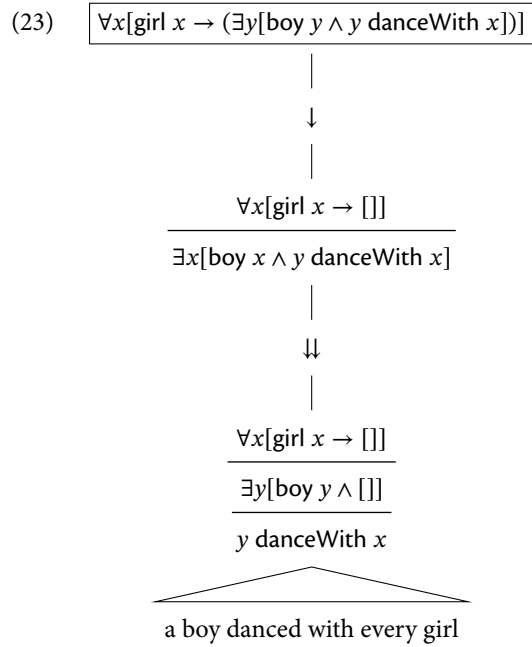
Great! We've shown how to achieve quantifier scope ambiguities using our new framework. Let's look at the derivations again side-by-side.

(24)   Surface scope (schematic derivation)

$\downarrow$
|
S
Q_1 S
$R^{\uparrow}$ Q_2
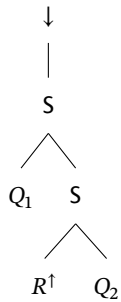
(25)   Inverse scope (schematic derivation)

$\downarrow$
|
$\Downarrow$
|
S_2
$Q_1^{\uparrow}$ S_2
$R^{\uparrow_2}$ $Q_2^{\uparrow\uparrow}$
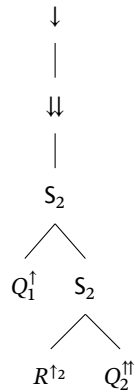
There's a couple of interesting things to note here:

- The inverse scope derivation involves more applications of our type-shifting operations – this becomes especially clear if we decompose the complex operations S_2, $\uparrow_2$, $\Uparrow$, and $\Downarrow$.

- In order to derive an inverse scope reading, what was *crucial* was the availability of *internal lift*; the remaining operations, S_2, $\uparrow_2$, $\Downarrow$ only functioned to

massage composition for three-story towers.

On the latter point, it's tempting to conjecture that in, e.g., German, Japanese and other languages which "wear their LF on their sleeve", the semantic correlate of *scrambling* is *internal lift*, whereas in scope-flexible languages such as English, internal lift is a freely available operation.[11]

If we adopt some version of the *derivational complexity hypothesis*[12], we also predict that inverse scope readings should take longer to process than surface scope readings.

It's worth mentioning, incidentally, that although we collapsed the resulting three-story tower via internal lower followed by lower, we can also define an operation that collapses a three-story tower two an ordinary tower in a different way. Let's call it *join*:[13]

(26)   *join* (def.)

$$m^{\mu} := \lambda k \,.\, m \,(\lambda c \,.\, c\, k) \qquad\qquad \mu : \mathsf{K_t}\,(\mathsf{K_t}\,\mathsf{a}) \to \mathsf{K_t}\,\mathsf{a}$$

In tower terms, join takes a three-story tower and sequences quantifiers from top to bottom:

(27) $$\left(\dfrac{\dfrac{f\,[\,]}{g\,[\,]}}{x}\right)^{\mu} = \dfrac{f\,(g\,[\,])}{x}$$

Doing internal lower on a three-story tower followed by lower is equivalent to doing join on a three-story tower followed by lower (as an exercise, convince yourself of this). However, there's may be a good empirical reason for having internal lower as a distinct operation (and since it's just lifted lower, it "comes for free" in a certain sense).

(28)   Daniele wants a boy to dance with every girl.   ∀ > want > ∃

Arguably, (28) can be true if for every girl x, Daniele has the following desire: *a boy dances with y*. This is the reading on which *every boy* scopes over the intensional verb, and *a boy* scopes below it.

If we have *internal lower*, getting this is easy. We *internally lift every girl* and *externally lift a boy*. Before we reach the intensional verb, we fix the scope of *a*

---

[11] To make sense of this, we would of course need to say something more concrete about overt movement. For an attempt at marrying continuations to a standard, minimalist syntactic component, see my manuscript *Movement as higher-order structure building*.

[12] I.e., that derivational complexity correlates with processing difficulty.

[13] Join for three-story towers corresponds directly to the *join* function associated with the continuation monad. For more on continuations from a categorical perspective, see the appendix to the first handout.

boy by doing internal lower. Now we have an ordinary tower, and we can defer
fixing the scope of *every girl* via *lower* until after the intensional verb.

(29)   Step 1: scope *every girl* over *a boy*

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\frac{\exists y[\text{boy } y \land []]}{y \text{ dance-with x}}}$$

S₂
├─ ...
│   └─ ↑
│   └─ a boy
└─ S₂
    ├─ dance with^{↑2}
    └─ ...
        └─ ↑↑
            └─ every girl
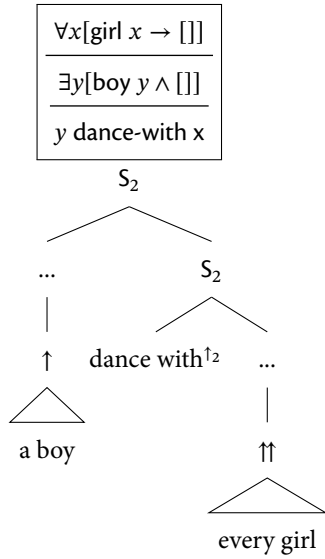
(30)   Step 2: internally lower below *want*

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\exists y[\text{boy } y \land y \text{ dance-with } x]}$$

⇓

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\frac{\exists y[\text{boy } y \land []]}{y \text{ dance-with x}}}$$
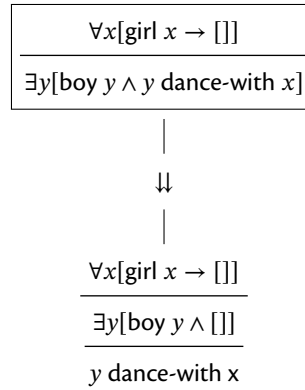
(31)   Step 3: scope *every girl* over *want*

$$\frac{\forall x[\text{girl } x \rightarrow []]}{\text{daniele want } (\exists y[\text{boy } y \land y \text{ dance-with } x])}$$

S
├─ Daniele^{↑}
└─ S
    ├─ wants^{↑}
    └─ $\frac{\forall x[\text{girl } x \rightarrow []]}{\exists y[\text{boy } y \land y \text{ dance-with } x]}$
        └─ ...

(32)   Step 4: lower

$$
\forall x \left[\begin{array}{l} \text{girl } x \\ \rightarrow \left(\text{daniele want} \left(\exists y \left[\begin{array}{l} \text{boy } y \\ \wedge\, y \text{ dance-with } x \end{array} \right] \right) \right) \end{array} \right]
$$

$$|$$

$$\downarrow$$

$$|$$

$$\dfrac{\forall x[\text{girl } x \rightarrow [\,]]}{\text{daniele want } (\exists y[\text{boy } y \wedge y \text{ dance-with } x])}$$
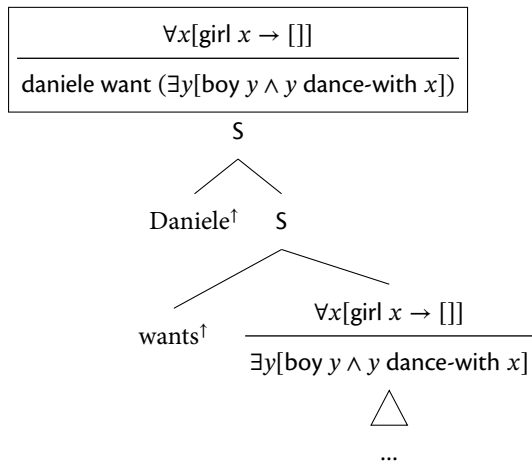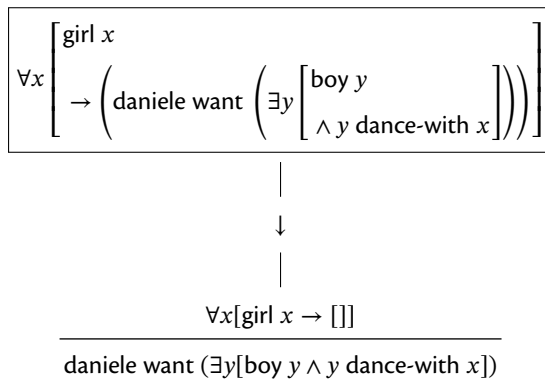
If we only have *join* then the scope of *a boy* and *every girl* may vary amongst themselves, but they should either both scope below *want* or both scope above *want*.

Note that we haven't given a concrete treatment of intensionality here, but, mechanically, we can simply replace every occurrence of t with type s → t. Intensional verbs take type s → t complements. Here are some sample lexical entries:

(33)   a. $[\![\text{every girl}]\!] := \lambda k \,.\, \lambda w \,.\, \forall x[\text{girl}_w\, x \rightarrow k\, x]$ $\qquad$ $(e \rightarrow (s \rightarrow t)) \rightarrow s \rightarrow t$
         b. $[\![\text{dance with}]\!] := \lambda yx \,.\, \lambda w \,.\, y \text{ dance-with}_w\, x$ $\qquad\qquad$ $e \rightarrow s \rightarrow t$
         c. $[\![\text{want}]\!] := \lambda p \,.\, \lambda x \,.\, \lambda w \,.\, x \text{ want}_w\, p$ $\qquad\qquad$ $(s \rightarrow t) \rightarrow e \rightarrow s \rightarrow t$

Instead of giving back a return type t, in an intensional setting our continuation type constructor is defined in terms of a return type s → t:

(34) $K_{s \rightarrow t}\, a := (a \rightarrow (s \rightarrow t)) \rightarrow s \rightarrow t$

You can verify for yourselves that the core features of the system remain unaffected, i.e, the definitions of lift and S can remain the same.

3   Split scope

In the first p-set, I asked you how to think about analyzing split scope of non-upward-monotone quantifiers:

(35)   The company need fire no employees.

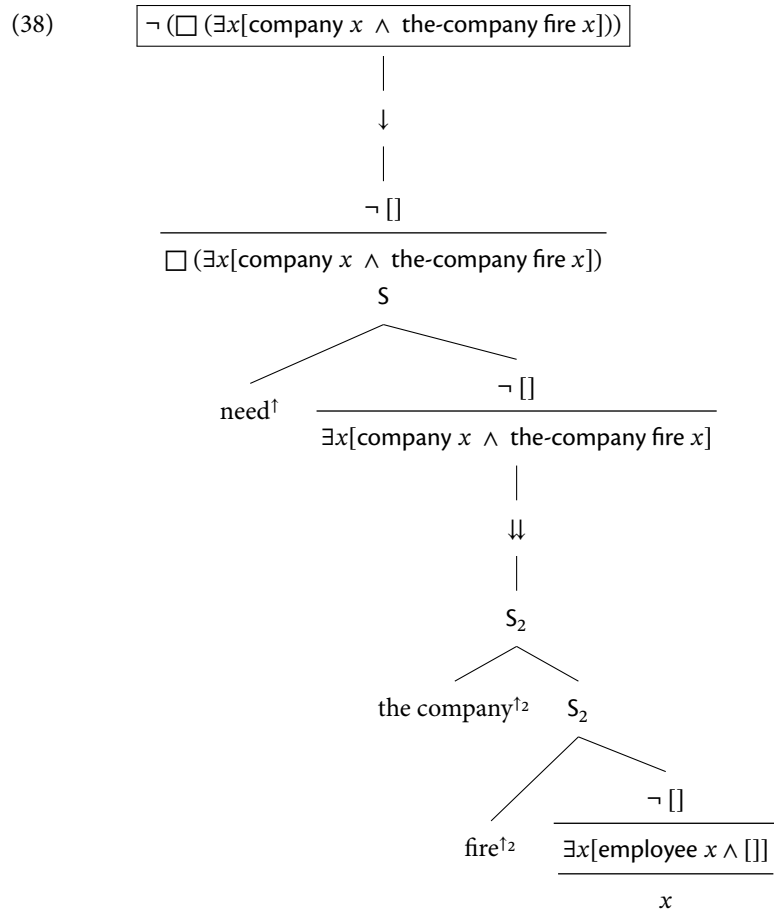*It's not the case that the company needs to hire employees.*   ¬ > □ > ∃

With continuation semantics, we can understand this data as providing support for the idea that expressions can denote three-story towers (something not excluded by, e.g., Heim & Kratzer 1998 in any case).

(36)   ⟦no employees⟧ := λk . ¬ k (λl . $\exists x$[employee $x \wedge l\ x$]) K_t (K_t a)

Tower version:

(37)   ⟦no employees⟧ :=
$$\frac{\neg\ []}{\exists x[\text{employee } x \wedge []]}$$
$$x$$

We get the split scope reading by doing internal lower first below the modal, and then external lower above the modal.

(38) $\boxed{\neg\ (\Box\ (\exists x[\text{company } x\ \wedge\ \text{the-company fire } x]))}$

|
↓
|

$$\frac{\neg\ []}{\Box\ (\exists x[\text{company } x\ \wedge\ \text{the-company fire } x])}$$
S

need↑ $$\frac{\neg\ []}{\exists x[\text{company } x\ \wedge\ \text{the-company fire } x]}$$

|
⇓
|

S_2

the company↑² S_2

fire↑² $$\frac{\neg\ []}{\exists x[\text{employee } x \wedge []]}$$
$$x$$

One interesting property of split scope readings is that it is essential that we be able to *lower* a 3-story tower in two distinct steps – *internal lower* followed by *lower*.

Despite being very elegant, I'm not sure whether this is a completely satisfactory account of split scope readings – it's been observed, for example, that split scope always seems to involve narrow scope of an existential (see, e.g., Abels & Martí 2010). This doesn't fall out from the analysis outlined here.[14]

[14] An investigation of split scope from the perspective of continuations could be a good project for this class – see, e.g., Bumford 2017 for relevant discussion.

4   Scope islands and obligatory evaluation

Quantifiers can't take scope arbitrarily high – rather, their scope is roofed by certain constituents. As is well known, the environments that are *islands for scope taking* don't necessarily correspond to the environments that are islands for overt movement (see May 1977).

We still need a theory of scope islands in order to restrict the power of continuation semantics. It turns out that there is a very natural notion available to us, similar to Chomsky's notion of a *phase*.

Inspired by research on *delimited control* in computer science[15], Charlow (2014) develops an interesting take on scope islands couched in terms of continuations.

[15] See, e.g., Danvy & Filinski 1992 and Wadler 1994.

He proposes the following definition:

(39)   *Scope islands* (def.)
    A *scope island* is a constituent that is subject to *obligatory evaluation*.
                           (Charlow 2014: p. 90)

By *obligatory evaluation*, we mean that every continuation argument *must* be saturated before semantic computation can proceed. In other words, a scope island is a constituent where, if we have something of type K_t a, we must lower it before we can proceed.

Let's be more precise:

(40)   a.   A constituent X is *evaluated* if it has an evaluated type a.
      b.   A type a is *evaluated* if a $\neq K_t$ b.

One way of thinking about this, is that the presence of an unsaturated contin-

uation argument means that there is some computation that is being deferred until later.

Scope islands are constituents at which evaluation is *forced*. As noted by Charlow, this idea bears an intriguing similarity to Chomsky's notion of a *phase*.[16]

[16] Exploring this parallel in greater depth could make for an interesting term paper topic.

How does this work in practice? A great deal of ink has been spilled arguing that, e.g., a finite clause is a scope island.

scope island

(41)   A boy said │ that Susan greeted every linguist │.   ∃ > ∀; ✗ ∀ > ∃

The derivation of the embedded clause proceeds as usual via lift and SFA.

(42)   Scope island with an unevaluated type
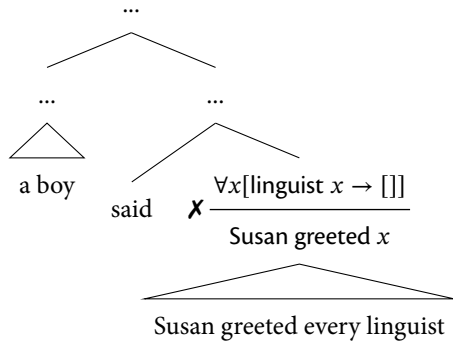
(43)   Scope island with an evaluated type
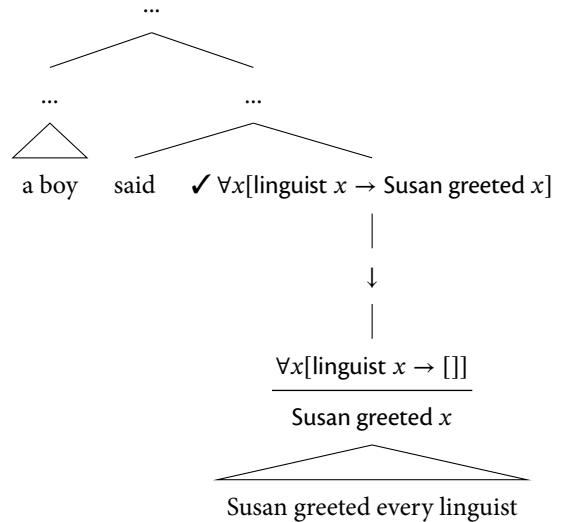
This story leaves a lot of questions unanswered of course:

- Is this just a recapitulation of a representational constraint on quantifier raising?[17]

- Are finite CPs the only scope island? What about DPs?[18]

- Can we give a principled story about islands for overt movement using similar mechanisms? What explains the difference between overt movement and scope taking with respect to locality?[19]

[17] The answer to this question may ultimate be *yes*, in my view.

[18] In my view, yes, the default assumption should be that DPs are scope islands. See Sauerland 2005 and Charlow 2010 for relevant discussion.

[19] If we want to give a more general account of phases using this mechanism, we need to give an account of overt movement in terms of continuations, too. See my unpublished ms. *Movement as higher-order structure building* for progress in this direction.

4.1   A brief remark on Scope Economy

As famously discovered by Fox 1995, scope-shifting operations are subject to an economy condition.

(44)   Economy condition on scope shifting (*Scope Economy*) (def.)
OP can apply only if it affects semantic interpretation (i.e., only if inverse-scope and surface-scope are semantically distinct).[20]

<div align="right">(Fox 2000: p.,21)</div>

[20] Here, OP stands for *scope-shifting operation*, i.e., quantifier raising.

Scope economy has an impressively wide empirical coverage, including interactions between scope and ellipsis. Consider, e.g., the following contrasts (from Fox 1995):

(45)   Some linguist likes every philosopher.   ∃ > ∀; ∀ > ∃

(46)   Some linguist likes every philosopher,
and some mathematician does too.   ∃ > ∀; ∀ > ∃

(47)   Some linguist likes every philosopher,
and Mary does too.   ∃ > ∀; ✗ ∀ > ∃

(48)   Some linguist likes every philosopher,
and every mathematician does too.   ∃ > ∀; ✗ ∀ > ∃

Scope economy gives us an explanation of this paradigm, just in case we assume that an elliptical sentence and its antecedent must involve *parallel scopal relations*.

In (48), for example, QR of *every philosopher* over *every mathematician* is blocked by scope economy, since the relative scope of two universal quantifiers doesn't affect truth-conditions.

(49) *every philosopher [λx every mathematician does $\boxed{\text{like } t_x}$]

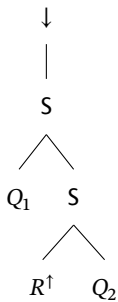Since the antecedent must involve a parallel scopal relation, inverse scope is blocked.

Here, I just want to point out that our continuation semantic framework is actually an extremely natural fit for scope economy:

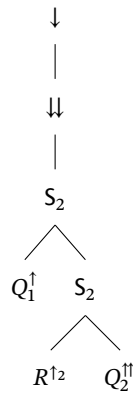(50)   Economy condition on scope shifting (continuations ver.)

> A derivation D is ruled out if there is a simpler derivation D' that gives rise to the same interpretation.

Just so long as the economy condition is evaluated at scope islands (as argued for by Fox 1995), we capture the basic observations, since scope *every philosopher* over *every mathematician*, involves, minimally, an additional internal lift of *every philosopher*.

(51)   Surface scope (schematic derivation)

$$
\begin{array}{c}
\downarrow \\
| \\
S \\
\diagup\diagdown \\
Q_1 \quad S \\
\diagup\diagdown \\
R^{\uparrow} \quad Q_2
\end{array}
$$

(52)   Inverse scope (schematic derivation)

$$
\begin{array}{c}
\downarrow \\
| \\
\Downarrow \\
| \\
S_2 \\
\diagup\diagdown \\
Q_1^{\uparrow} \quad S_2 \\
\diagup\diagdown \\
R^{\uparrow 2} \quad Q_2^{\uparrow\uparrow}
\end{array}
$$

It remains to be seen whether an economy condition framed in terms of the complexity of a continuation-semantic derivation makes any distinct predictions to an economy condition framed in terms of QR.[21]

[21] Figuring this out would be a great student project, I think.

5   Generalized con/dis-junction

The flexibility of *and* and *or* has been discussed at length by, e.g., Partee & Rooth (1983), Winter (2001), among others.[22]

[22] We saw an example of this when we motivated LIFT.

(53)   Lan and some woman arrived.

(54)   Howie sneezed and/or coughed.

(55)   Lan kissed and/or hugged Irene.

Unlike other expressions we've seen so far, we can characterize *and* and *or* as expressions that takes two continuized values as arguments.[23]
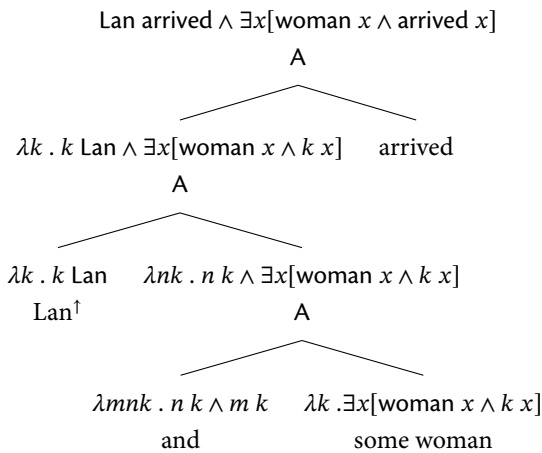
[23] Note that since the continuation argument k occurs more than once in the function body, we can no longer abbreviate the flat lambda-expression using a tower.

(56)   a.   *m* and *n* := $\lambda k \,.\, m\,k \,\wedge\, n\,k$ and : $K_t\,a \rightarrow K_t\,a \rightarrow K_t\,a$

     b.   *m* or *n* := $\lambda k \,.\, m\,k \,\vee\, n\,k$ or : $K_t\,a \rightarrow K_t\,a \rightarrow K_t\,a$

The intuition here is as follows: *and* wants as its arguments things that are *guaranteed* to give back truth values at some future stage of computation.

This accounts for the basic cases discussed by Partee & Rooth (1983), with a single (polymorphic) entry for *and*, as illustrated below:[24]
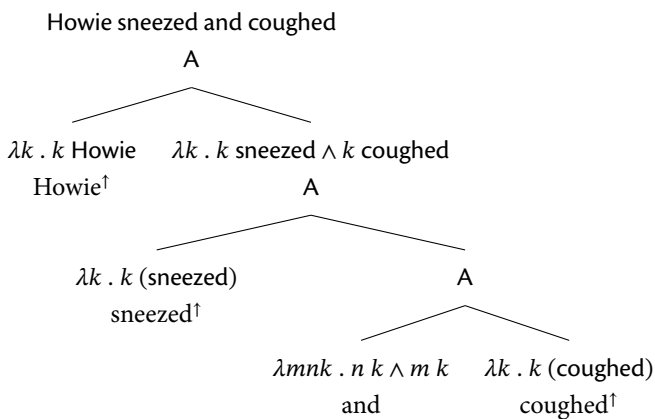
(57)   Lan and some woman arrived.

$$\text{Lan arrived} \wedge \exists x[\text{woman } x \wedge \text{arrived } x]$$
$$\text{A}$$

$$\lambda k \,.\, k \text{ Lan} \wedge \exists x[\text{woman } x \wedge k\, x] \qquad \text{arrived}$$
$$\text{A}$$

$$\lambda k \,.\, k \text{ Lan} \qquad \lambda n k \,.\, n\,k \wedge \exists x[\text{woman } x \wedge k\,x]$$
$$\text{Lan}^{\uparrow} \qquad\qquad \text{A}$$

$$\lambda m n k \,.\, n\,k \wedge m\,k \qquad \lambda k\,.\exists x[\text{woman } x \wedge k\,x]$$
$$\text{and} \qquad\qquad \text{some woman}$$

(58)   Howie sneezed and coughed.

$$\text{Howie sneezed and coughed}$$
$$\text{A}$$

$$\lambda k \,.\, k \text{ Howie} \qquad \lambda k \,.\, k \text{ sneezed} \wedge k \text{ coughed}$$
$$\text{Howie}^{\uparrow} \qquad\qquad \text{A}$$

$$\lambda k \,.\, k \text{ (sneezed)} \qquad\qquad \text{A}$$
$$\text{sneezed}^{\uparrow}$$

$$\lambda m n k \,.\, n\,k \wedge m\,k \qquad \lambda k \,.\, k \text{ (coughed)}$$
$$\text{and} \qquad\qquad \text{coughed}^{\uparrow}$$

5.1   *The scope of con/dis-junction*

In a lot of the cases we've seen so far, teasing apart the power of QR from the power of continuations hasn't been at all trivial. In this section, we'll see an

[24] Unlike other lexical entries we've seen so far, *and* is *lexically specified* as seeking continuized arguments. As such, if a value isn't already typed as an instantiation of $K_t\,a$, it must be lifted.

In the derivations below, you can observe that, unlike the cases we've encountered so far, *and* composes with its arguments via Function Application (FA) rather than sFA.

example of a case where it's clearer that continuations can help us in ways that QR can't.

Both conjunction and disjunction exhibit "scope" ambiguities. This is illustrated below for conjunction:

(59)   You're not allowed to dance and sing.
     a.   *You're not allowed to dance and you're not allowed to sing*

$$\land > \lnot > \Diamond$$

     b.   *You're not allowed to both dance and sing (at the same time)*

$$\lnot > \Diamond > \land$$

We can account for the wide/narrow scope ambiguity as a matter of where we LOWER.

Let's first compute the semantic value of the prejacent of the modal:

(60) $\lambda k \;.\; k \;(\text{you dance}) \land k \;(\text{you sing})$
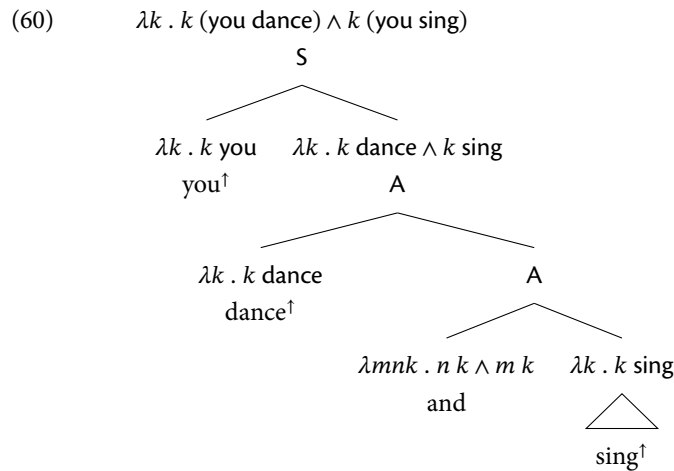
If we LOWER immediately we're just going to get a proposition, which allowed will take as its argument, deriving the narrow scope reading.[25]

The "wide scope" reading is more interesting. We can simply defer lowering, and compose the prejacent with lifted *allowed* via S.

[25] Instead of composing via S and lowering, we could equivalently compose lifted *you* with its complement via A, since it's a subject.

(61) $(\neg (\Diamond \text{ (you sing)})) \wedge (\neg (\Diamond \text{ (you dance)}))$

|

LOWER

|

$\lambda k \,.\, k \,(\neg (\Diamond \text{ (you sing)})) \wedge k \,(\neg (\Diamond \text{ (you dance)}))$
S

$\lambda k \,.\, k \,(\lambda p \,.\, \neg \, p)$ S
not$^{\uparrow}$

$\lambda k \,.\, k \,(\lambda p \,.\, \Diamond \, p)$ $\lambda k \,.\, k \text{ (you dance)} \wedge k \text{ (you sing)}$
allowed$^{\uparrow}$

you dance and sing
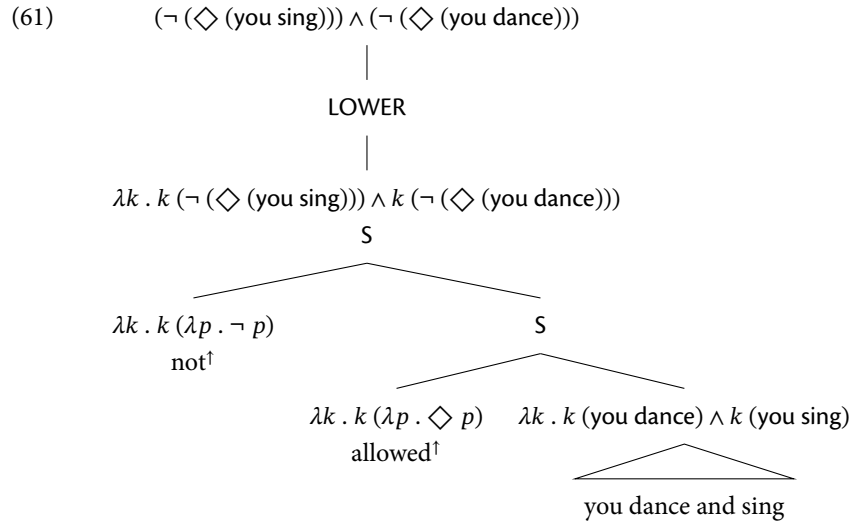
We make the nice prediction that "wide scope" readings of conjunction should be subject to scope islands:

(62)   John isn't allowed to claim [that you sing and dance].   ✗ $\wedge > \neg > \Diamond$

5.2   *Split scope with conjunction*

Hirsch (2017) claims that conjunction reduction is necessary in order to derive the "split scope" reading of the following sentence:

(63)   John refused to visit any city in Europe and any city in Asia.
       John refused to visit any city in Europe, and he refused to visit any city in
       Asia. $\wedge > \text{refuse} > \text{any}$

Hirsch's analysis:

(64)   John [$_{vP}$ refused to visit any city in Europe] and
       [$_{vP}$ refused to visit any city in Asia]

Here, I'm going to argue that we can get the split scope reading of (63) using just the machinery we've already introduced. Concretely:

- Our entry for *and*.

- The free availability of *external lift*.

First of all, each conjunct is going to be *externally lifted*.

(65)   ⟦any city in Europe⟧ $^\uparrow$ = $\lambda l . l (\lambda k . \exists x[\text{city-europe } x \wedge k\ x])$
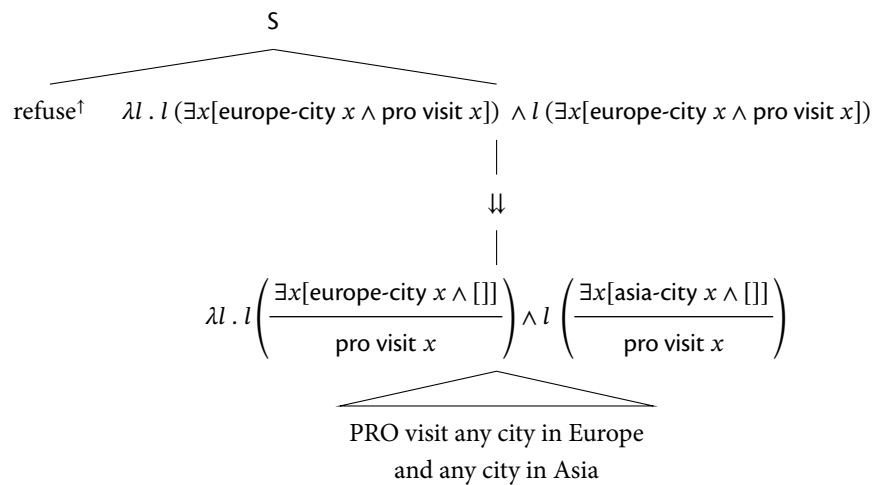
and, in our view, is polymorphic – it's looking for two arguments of type K_t a.
This means it can compose the externally lifted DPs, returning the following
meaning:

(66) $\lambda l . l \left(\dfrac{\exists x[\text{europe-city } x \wedge []]}{x} \right) \wedge l \left(\dfrac{\exists x[\text{asia-city } x \wedge []]}{x} \right) : K_t (K_t \text{ a})$

Remember that S_2 is just like S, only it does S of the wrapped values. Composition can proceed via $\uparrow_2$ and S_2 up to the prejacent of *refuse*.

To get the scope of the quantifiers right, we do *internal lower* just before we
compose with *refuse*.

(67) $\lambda l . l$ (refuse $\exists x[\text{europe-city } x \wedge$ pro visit $x]) \wedge l$ (refuse $\exists x[\text{europe-city } x \wedge$ pro visit $x])$

S

refuse$^\uparrow$ $\lambda l . l$ ($\exists x[\text{europe-city } x \wedge$ pro visit $x]) \wedge l$ ($\exists x[\text{europe-city } x \wedge$ pro visit $x])$

⇓

$\lambda l . l \left(\dfrac{\exists x[\text{europe-city } x \wedge []]}{\text{pro visit } x} \right) \wedge l \left(\dfrac{\exists x[\text{asia-city } x \wedge []]}{\text{pro visit } x} \right)$

PRO visit any city in Europe
and any city in Asia

Lowering the result gives us...the split-scope reading:

(68)   refuse $\exists x[\text{europe-city } x \wedge$ pro visit $x \wedge$ refuse $\exists x[\text{asia-city } x \wedge$ pro visit $x]$

LOWER

$\lambda l . l$ (refuse $\exists x[\text{europe-city } x \wedge$ pro visit $x]) \wedge l$ (refuse $\exists x[\text{europe-city } x \wedge$ pro visit $x])$

So, continuations can get split-scope readings of conjunction straightforwardly. This is a notable result.[26]

Note that this theory of split-scope coordination explains the lack of split-scope reading in the following example, from Partee & Rooth (1983).

(69)   John hopes [that some company will hire a maid and a cook].
        ✗ *John hopes that some company will hire a maid,*
        and John hopes that some company will hire a cook.

Hirsch doesn't consider an analysis of this data in terms of continuations, but does entertain a similar analysis involving lifting the quantificational conjuncts, alongside QR with higher-type traces. Hirsch (2017) rejects this analysis on the basis that it allows for the quantifiers to be syntactically above the intensional verb, while semantically reconstructing below it – a configuration which has been argued to be ruled out. Note that an analysis in terms of continuations doesn't face this objection – at no point in the analysis did we need to invoke covert movement.

This is simply because the embedded finite clause is a scope island, and therefore the scope of *and* is trapped.

6   Extension i: DP internal composition and indexed continuations

As you'll probably have noticed, we've spent this whole time treating quantificational DPs such as *every boy* as primitives.

At this point a natural question to ask is: how do determiners compose with their restrictors?

Surprisingly, the answer isn't as straightforward as you might think.

Naively, we may assume that determiners receive they're standard meaning – essentially, a function from a predicate to a *continuized* individual.

(70) $\llbracket \text{every} \rrbracket \overset{?}{:=} \lambda P . \dfrac{\forall y [P\ y \rightarrow []]}{y}$

This will (obviously) work fine for a nominal restrictor.

But, what happens if the restrictor itself contains a quantificational expression? Consider the following example:
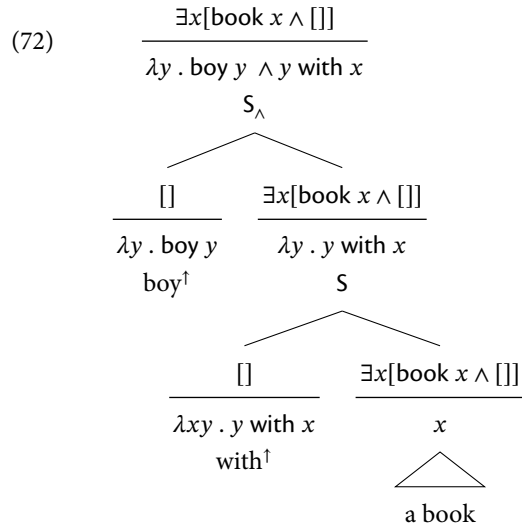
(71)   Every boy with a book left. $\forall > \exists$

Let's first compute the meaning of the restrictor *boy with a book*:[27]

(72)

$$\frac{\exists x[\text{book } x \wedge []]}{\lambda y . \text{ boy } y \ \wedge y \text{ with } x}$$

$$S_\wedge$$

$$\frac{[]}{\lambda y . \text{ boy } y}$$
$$\text{boy}^\uparrow$$

$$\frac{\exists x[\text{book } x \wedge []]}{\lambda y . \ y \text{ with } x}$$
$$S$$

$$\frac{[]}{\lambda xy . \ y \text{ with } x}$$
$$\text{with}^\uparrow$$

$$\frac{\exists x[\text{book } x \wedge []]}{x}$$

a book

What we end of with is a continuized predicate of type $\frac{t}{e \to t}$.

How do we compose this with our determiner of type $(e \to t) \to \frac{t}{e}$?

One possiblity is to simply lift the determiner. There are two problems with this approach.

- This will give *a book* scope over *every* – here, we're interested in the surface scope reading.

- Despite potentially being a useful strategy for deriving inversely-linked interpretations, this will ultimately allow the inner quantifier to take scope outside of the containing DP – this runs into issues with Larson's generalization. We'll come back to this.[28]

Instead, we're going to pursue the idea that the determiner itself *takes scope*.[29]

Consider our type constructor K_t – it takes a type a and returns a new type $(a \to t) \to t$.

In principle, we could parameterize K to any type r (here $r = t$). Let's call r the *return type*, since it tells us the type of the value we get when the continuation argument k applies to its argument.

[28] Essentially, there is some quite strong evidence that DP is a scope island.

[29] This has many precedents in the literature. See, for example, **heim1982**.

What if applying k to its argument gives back an intermediate result of type $\mathtt{i}$, which is subsequently transformed into a final result of type $\mathtt{r}$? We can model this idea using the type constructor $\mathsf{K}_{\mathtt{r}}^{\mathtt{i}}$:[30]

(73) $\mathsf{K}_{\mathtt{r}}^{\mathtt{i}}\, \mathtt{a} := (\mathtt{a} \to \mathtt{i}) \to \mathtt{r}$

[30] This ultimately goes back to Wadler 1994.

Barker & Shan (2014) generalize tower notation to the more general type-schema.[31]

[31] See also Charlow (2014: chapter 3).

(74)   Tripartite tower types (def.)

$$\frac{\mathtt{r} \,\big|\, \mathtt{i}}{\mathtt{a}} := (\mathtt{a} \to \mathtt{i}) \to \mathtt{r}$$

We can think of our existing tower notation as an abbreviation for a tripartite tower type, where the intermediate and final result types happen to be the same:

(75)   Bipartite towers as abbreviations for tripartite towers

$$\frac{\mathtt{r}}{\mathtt{a}} := \frac{\mathtt{r} \,\big|\, \mathtt{r}}{\mathtt{a}}$$

Now that we have tripartite tower types, we can think of the restrictor argument c of *every* as a *continuation argument*.

(76)   Standard determiner semantics for *every*

$$[\![\text{every}]\!] := \lambda c \, . \, \left[\lambda P \, . \, \frac{\forall y[P\, y \to []]}{y} \right] (\lambda x \, . \, c\, x) \qquad\qquad (\mathtt{e} \to \mathtt{t}) \to \frac{\mathtt{t}}{\mathtt{e}}$$

We can abbreviate the meaning of *every* as a tower, where c is the continuation argument:

(77) $$\frac{\left[\lambda P \, . \, \dfrac{\forall y[P\, y \to []]}{y} \right] (\lambda x \, . \, [])}{x} \qquad\qquad : \; \frac{\dfrac{\mathtt{t}}{\mathtt{e}} \,\bigg|\, \mathtt{t}}{\mathtt{e}}$$

Our existing definition of S can be made more type-general, in order to accommodate tripartite tower types. *Adjacent types* match and cancel out:

(78) $S : \dfrac{\dfrac{r\,|\,i}{a \to b} \quad \dfrac{i\,|\,j}{a}}{\dfrac{r\,|\,j}{b}}$

The actual definition of S doesn't change.

(79)   *scopal function application* (def.)
$m \ S \ n := \lambda k . m (\lambda x . n (\lambda y . k (x \ A \ y)))$

Likewise, the type of *lower* is further generalized; the definition doesn't change:

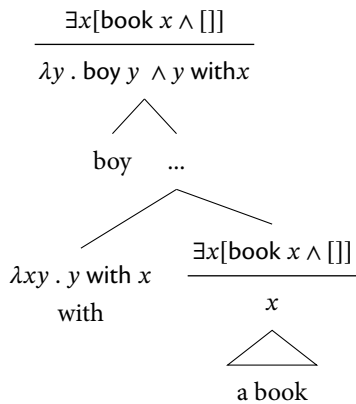(80) $(\downarrow) : \dfrac{\dfrac{a\,|\,b}{b}}{} \to a$

We can now make sense of the idea that the determiner *takes scope*. Let's consider again the following complex DP:

(81)   Every boy with a book.

First, we compute the meaning of the restrictor, returning something of type $\dfrac{t\,|\,t}{e \to t}$:

(82)   Step 1: compute *boy with a book*

$$\dfrac{\exists x[\text{book } x \wedge [\,]]}{\lambda y . \text{boy } y \ \wedge y \text{ with} x}$$

boy   ...

$\lambda xy . y$ with x $\dfrac{\exists x[\text{book } x \wedge [\,]]}{x}$

with

a book

Recall that *every* is of type $\dfrac{\dfrac{t}{e}\bigg|\,t}{e}$, this is something that can compose with the

restrictor via our more type-general formulation of sfa.

(83)

$$\frac{\forall y[(\exists x[\text{book } x \wedge \text{boy } y \wedge y \text{ with } x]) \rightarrow []]}{y}$$

|

$$\left[\lambda P . \frac{\forall y[P\ y \rightarrow []]}{y}\right] (\lambda y . \exists x[\text{book } x \wedge \text{boy } y \wedge y \text{ with } x])$$

|

↓

|

$$\frac{\left[\lambda P . \dfrac{\forall y[P\ y \rightarrow []]}{y}\right] (\lambda y . \exists x[\text{book } x \wedge []])}{\text{boy } y \wedge y \text{ with } x}$$

S

$$\frac{\left[\lambda P . \dfrac{\forall y[P\ y \rightarrow []]}{y}\right] (\lambda y . [])}{y} \qquad \frac{\exists x[\text{book } x \wedge []]}{\lambda y . \text{boy } y \wedge y \text{ with } x}$$

boy with a book
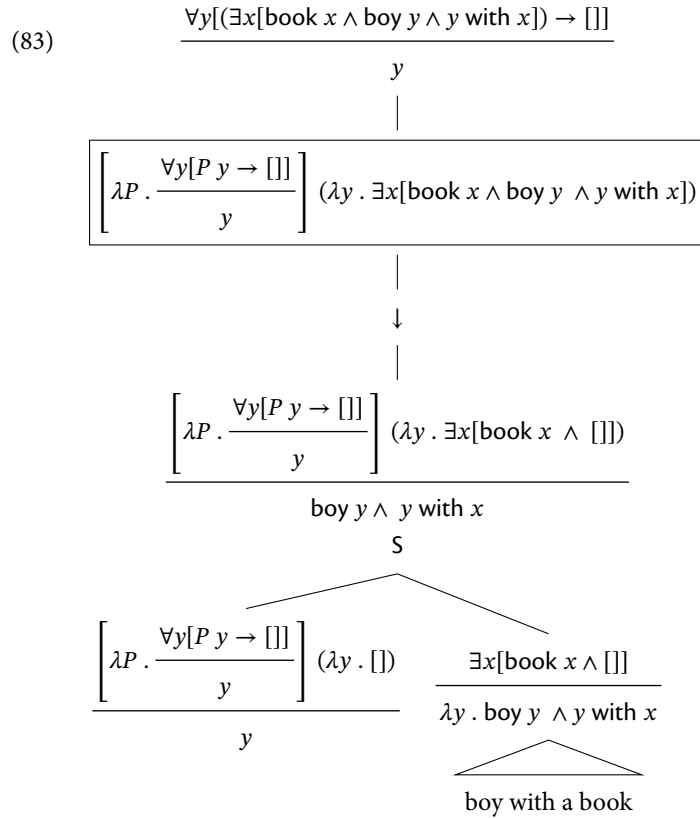
Exercise

- -

As homework, you can attempt to work out the above derivation yourself. Convince yourself that the types work out.

---

7   Extension ii: Exceptional scope of indefinites

Unlike other quantificational expressions, indefinites don't obey scope islands:

(84)   Roger hopes that some student will be absent.   ∃ > hope

Charlow (2014) outlines a theory of exceptional scope framed in terms of continuation semantics, compatible with the assumption that scope islands are obligatorily evaluated.

Charlow assumes that, unlike other quantificational expressions, indefinites

denote sets of individuals:

(85)   a. $[\![$some student$]\!] := \{\, x \mid \text{student } x \,\}$
        b. $[\![$some student$]\!] : \{\, e \,\}$

The central component of Charlow's proposal is a method (let's call it $\star$) for lifting a set into a scope taker.[32]

(86) $m^{\star} := \lambda k . \bigcup_{x \in m} k\, x$ $\qquad\qquad \star : \{\, e \,\} \to (e \to \{\, t \,\}) \to \{\, t \,\}$

In fact, we can write $\star$ using tower notation:

(87) $m^{\star} := \dfrac{\displaystyle\bigcup_{x \in m}\ [\,]}{x}$

As you'll notice, $\star$ gives back a *continuized individual*, but with the return type generalized to $\{\, t \,\}$ rather than t.

Our existing definitions of lift and SFA allow us to compose a $\star$-shifted indefinite straightforwardly, once the return type is set to $\{\, t \,\}$. The actual definitions of the operations don't change.

Here I'll show how $\star$, once combined with our existing mechanisms for doing scope-taking – namely lift and S – *predicts* that indefinites can exceptionally scope out of scope islands.

I'll demonstrate this for our example: *Roger hopes that some student is absent*:
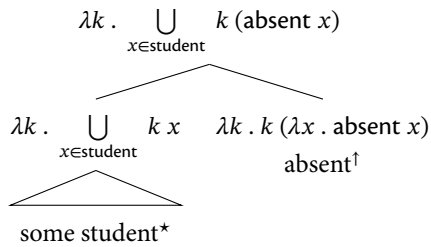
(88)   Step 1: lift *some student* into a scope-taker

$\lambda k . \bigcup_{x \in \text{student}} k\, x$

$|$

$\star$

$|$

$\{\, x \mid \text{student } x \,\}$
    some student

(89)   Step 2: scope via lift and S

$$\lambda k \; . \; \bigcup_{x \in \text{student}} k \; (\text{absent } x)$$

$$\lambda k \; . \; \bigcup_{x \in \text{student}} k \; x \qquad \lambda k \; . \; k \; (\lambda x \; . \; \text{absent } x)$$
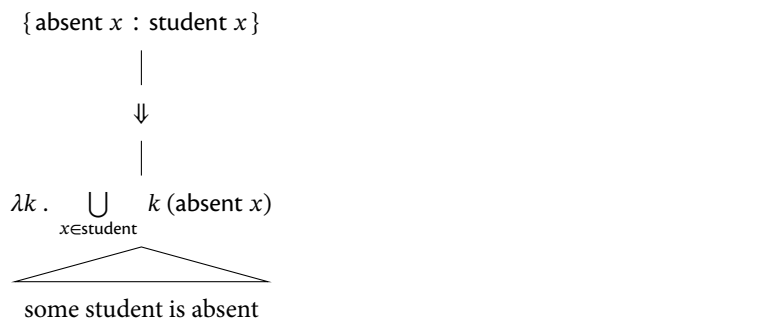
$$\text{absent}^\uparrow$$

some student*

We've reached a scope island now, so we need to *evaluate* it. We can't use our ordinary lower function – since we're dealing with sets, we lower by feeding n Partee's IDENT, rather than the identity function:

(90) $m^\Downarrow := m \; (\lambda x \; . \; \{ x \})$ $\qquad\qquad\qquad$ $\Downarrow : ((a \to \{ a \}) \to \{ a \}) \to \{ a \}$

(91)   Step 3: evaluate the scope island:

$$\{ \text{absent } x \; : \; \text{student } x \}$$

$$\Downarrow$$

$$\lambda k \; . \; \bigcup_{x \in \text{student}} k \; (\text{absent } x)$$

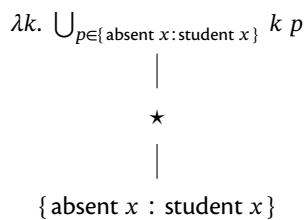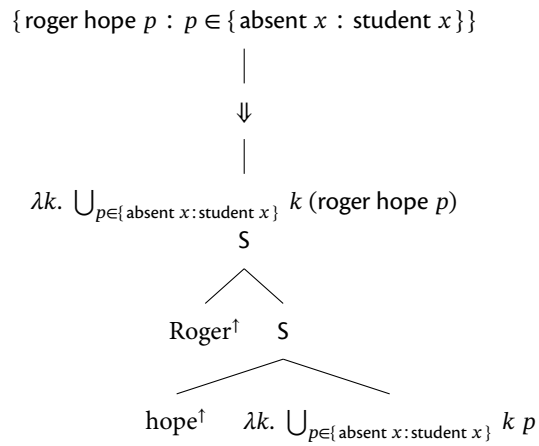some student is absent

What we end up with is a set of propositions of the form *x is absent*, where x is a student.

Now we re-lift the scope island into a scope taker via $\star$:

(92)   Step 4: re-lift via $\star$

$$\lambda k . \bigcup_{p \in \{ \text{absent } x \, : \, \text{student } x \}} k \; p$$

$$\star$$

$$\{ \text{absent } x \; : \; \text{student } x \}$$

Now we scope out the scope-island via lift and S – the indeterminacy introduced by the indefinite survives, and we get exceptional scope.
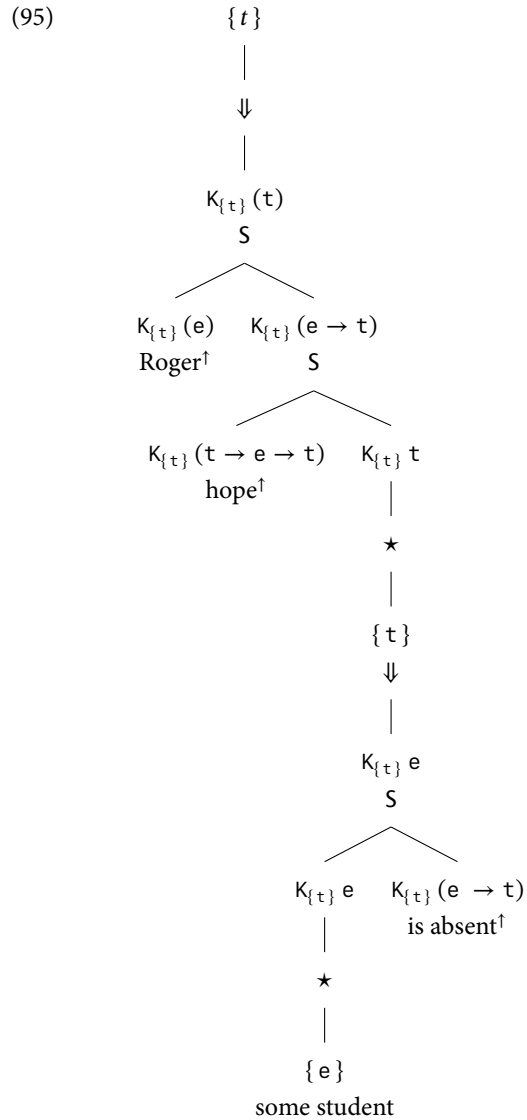
(93)   Step 5: scope via lift and $\star$:

$\{\, \text{roger hope } p \,:\, p \in \{\, \text{absent } x \,:\, \text{student } x \,\}\,\}$

|

$\Downarrow$

|

$\lambda k. \; \bigcup_{p \in \{\, \text{absent } x \,:\, \text{student } x \,\}} k \; (\text{roger hope } p)$
S

Roger$^\uparrow$ S

hope$^\uparrow$ $\lambda k. \; \bigcup_{p \in \{\, \text{absent } x \,:\, \text{student } x \,\}} k \; p$

The result of lowering is equivalent to the following:

(94) $\{\, \text{roger hope (absent x)} : \text{student x} \,\}$

In brief, unlike other quantifiers, indefinites introduce indeterminacy (which we model as a set). Indeterminacy survives obligatory evaluation at a scope island.

Here is the derivation again from a bird's eye view:

(95) $\{t\}$

$\Downarrow$

$\mathsf{K}_{\{t\}}(t)$
S

$\mathsf{K}_{\{t\}}(e)$ $\mathsf{K}_{\{t\}}(e \to t)$
Roger$^\uparrow$ S

$\mathsf{K}_{\{t\}}(t \to e \to t)$ $\mathsf{K}_{\{t\}}\,t$
hope$^\uparrow$

$\star$

$\{t\}$
$\Downarrow$

$\mathsf{K}_{\{t\}}\,e$
S

$\mathsf{K}_{\{t\}}\,e$ $\mathsf{K}_{\{t\}}(e \to t)$
                          is absent$^\uparrow$

$\star$

$\{e\}$
some student

References

Abels, Klaus & Luisa Martí. 2010. A unified approach to split scope. *Natural Language Semantics* 18(4). 435–470.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language* (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Bumford, Dylan. 2017. Split-scope definites: Relative superlatives and haddock descriptions. *Linguistics and Philosophy* (40). 549–593.

Charlow, Simon. 2010. Can DP Be a Scope Island? In Thomas Icard & Reinhard Muskens (eds.), *Interfaces: Explorations in Logic, Language and Computation* (Lecture Notes in Computer Science), 1–12. Berlin, Heidelberg: Springer.

Charlow, Simon. 2014. *On the semantics of exceptional scope.*

Chomsky, Noam. 1995. *The Minimalist program* (Current Studies in Linguistics 28). Cambridge Massachussetts: The MIT Press. 420 pp.

Chomsky, Noam. 2001. Derivation by phase. In Kenneth L. Hale & Michael J. Kenstowicz (eds.), *Ken hale: A life in language* (Current Studies in Linguistics 36). Cambridge Massachussetts: The MIT Press.

Danvy, Oliver & Andrzex Filinski. 1992. Representing Control: a Study of the CPS Transformation. *Mathematical Structures in Computer Science* 2(4). 361–391.

Elliott, Patrick D. 2019. Overt movement as higher-order structure building. unpublished manuscript. Leibniz-Zentrum Allgemeine Sprachwissenschaft.

Fox, Danny. 1995. Economy and scope. *Natural Language Semantics* 3(3). 283–341.

Fox, Danny. 2000. *Economy and semantic interpretation* (Linguistic Inquiry Monographs 35). Cambridge, Mass: MIT Press. 215 pp.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar* (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.

Hirsch, Aron. 2017. A case for conjunction reduction. unpublished manuscript. Massachussetts Institute of Technology.

Kobele, Gregory. 2006. *Generating copies - An investigation into structural identity in language and grammar.* UCLA dissertation.

May, Robert. 1977. *The grammar of quantification.* Massachussetts Institute of Technology dissertation.

Montague, Richard. 1973. The Proper Treatment of Quantification in Ordinary English. In K. J. J. Hintikka, J. M. E. Moravcsik & P. Suppes (eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics* (Synthese Library), 221–242. Dordrecht: Springer Netherlands.

Partee, Barbara & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In, Reprint 2012, 361–383. Berlin, Boston: De Gruyter.

Sauerland, Uli. 2005. DP is Not a Scope Island. *Linguistic Inquiry* 36(2). 303–314.

Stabler, Edward. 1997. Derivational Minimalism. In, 68–95.

Wadler, Philip. 1994. Monads and composable continuations. *LISP and Symbolic Computation* 7(1). 39–55.

Winter, Yoad. 2001. *Flexibility principles in boolean semantics: The interpretation of coordination, plurality, and scope in natural language* (Current Studies in Linguistics 37). Cambridge Massachussetts: The MIT Press. 297 pp.